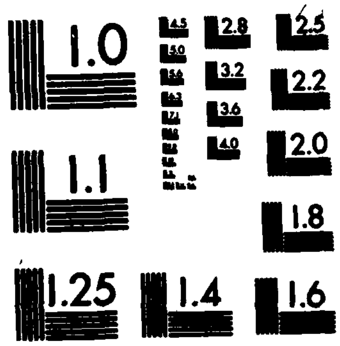
AD-A127 424 AN INTERACTIVE COMPUTER CODE FOR WATER WAVES IN OPEN CHANNELS(U) NAVAL RESEARCH LAB WASHINGTON DC N C CHU ET AL. 13 APR 83 NRL-MR-5066 1/1

UNCLASSIFIED F/G 9/2 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

NRL Memorandum Report 5066

AD A127424

An Interactive Computer Code for Water Waves in Open Channels

C. CHU AND G. A. KERAMIDAS

Fluid Dynamics Branch
Marine Technology Division

April 13, 1983

NAVAL RESEARCH LABORATORY
Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 5066	2. GOVT ACCESSION NO. AD-A127 424	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN INTERACTIVE COMPUTER CODE FOR WATER WAVES IN OPEN CHANNELS		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing problem
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) N. C. Chu and G. A. Keramidas		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, DC 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61153N23, RR0230141 1470-00
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, VA 22217		12. REPORT DATE April 13, 1983
		13. NUMBER OF PAGES 32
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Finite elements
Computer codes
Water waves
Shallow water equations

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A computational model has been developed in the Fluid Dynamics Branch of the Naval Research Laboratory for the simulation of water waves in open channels. The model numerically solves a set of partial differential equations known as the Shallow Water equations. The method employed for the solution of these equations is the Finite Element Method with linear element approximations in time and space.

The computer code was initially developed for the computational model on a HP-1000 computer system. This report contains the interactive version of the computer code, its description and provides information on how the code can be used.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

COMPUTATIONAL HYDRODYNAMICS: AN INTERACTIVE COMPUTER CODE FOR WATER WAVES IN OPEN CHANNELS

I. INTRODUCTION

The analysis and solution of free surface wave problems requires the development of a model which can represent the physical phenomenon under study. However the complexity of the equations describing surface waves imposes certain restrictions in the type of solution that one can use. Most often these equations are solved by some approximate method which requires the development of a computational model implementing such a method.

For the purpose of studying water waves in open channels a computational model has been developed. The analytical formulation and the description of the model can be found in [1]. The one-dimensional form of the shallow water equations is solved numerically by the finite element method (FEM). The simplest form of approximation in space and time is utilized for the FEM and the derived model is capable of producing accurate solutions to a variety of problems with free surface motion [2,3].

The numerical solution of the system of equations which represent the computational model is obtained by a computer code developed on an HP 1000 computer. A modification of the code was necessary in order to utilize the code in an interactive way and to be able to solve different types of cases with the least amount of user effort. Following a brief description of the analytical model in the first part of this report, the computer model is presented in the second part. The different parts of the computer model are given in a flow chart diagram and for each part of the required subroutines are described. In the last part of the report the solutions of some example problems are given. A user's manual and a listing of the computer code are included in Appendices A and B respectively.

II. COMPUTATIONAL MODEL

1. Analytical Model

The motion of a body of water with a free surface in a channel can be described by a set of two nonlinear partial differential equations (shallow water equations). The one dimensional form of these equations is

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(VH) = 0 \tag{1}$$

$$\frac{\partial V}{\partial t} + V\frac{\partial V}{\partial x} + g\frac{\partial h}{\partial x} = 0 \tag{2}$$

where h is the instantaneous water surface elevation from the mean surface level, H is the total depth of the fluid, V is the depth-averaged velocity in the x-direction, t is the time and g is the gravitational acceleration. Equations (1) and (2) are cast into a variational form and the FEM is applied to derive a system of ordinary differential equations given by

$$[M]\{\dot{q}\} + [K]\{q\} = \{Q\} \tag{3}$$

Manuscript approved February 25, 1983.

where $[M]$ is the mass matrix, $[K]$ is the convective matrix, $\{Q\}$ is a vector representing boundary forces and $\{q\}$ is the vector representing either $\{h\}$ or $\{V\}$.

The system of equations represented by Eq. (3) is reduced to an algebraic system by approximating the time derivative and the system then is given by

$$[A]\{q^{n+1}\} = \{Q\} + [B]\{q^n\} \tag{4}$$

where the matrices $[A]$ and $[B]$ are combinations of the matrices $[M]$ and $[K]$. The vectors $\{q^{n+1}\}$ and $\{q^n\}$ represent the unknowns at time levels $(n + 1)$ and (n) respectively. The derivation of the above equations can be found in [1].

The system of equations represented by Eq. (4) is solved numerically by a computer code developed at NRL for that purpose. The different parts of the code are described in the following section.

2. Computer Model

The analytical model described in the previous section can be used to solve a variety of physical problems concerned with water waves. Solutions are obtained by casting the analytical model into a computer model. A general description of the computer model is given in the flow chart, Fig. 1. The different stages of the model generate the required information and perform the numerical solution for the specified problem.

In the first stage the physical problem under consideration is defined. This includes the particular channel geometry, boundary conditions and initialization of all program parameters. The subroutines called in this stage are the INPUT and GEOMT subroutines. In the next stage the system of algebraic equations is formed (subroutine ELMAT) and the system is solved (subroutine QSOLV) for the particular type of boundary conditions under consideration (subroutines MPART, DCOMP, and SOLVE). Following the solution, the values for the surface elevation, velocity and pressure are computed. The process is repeated for every time increment. The last stage of the code performs such tasks as output of results, plotting of results either at specified time intervals or at the end of the solution. This stage includes subroutines XOUT, XPLOT, and TOUT, TPLOT.

As was mentioned earlier the noninteractive version of the computer code can be used on any computer. The interactive version of the code utilizes the HP-1000 graphics terminal 2648 and certain changes will be necessary in order for the code to be used on another machine.

The code has been optimized for minimum storage requirements and execution time. Further optimization can be achieved by utilizing the vector instruction set (VIS) of the HP-1000 computer system. Although this reduces the execution time, it also restricts the transferability of the code to other machines.

The description of each of the subroutines is given in the next section.

III. COMPUTER PROGRAM

3. Program Description

The computer program is composed by the main program (program KYMA) and eleven subroutines. The dimensions of the arrays are set for a maximum of 222 equations. This allows for 111 nodal points and two unknowns per point. The storage requirements for the set dimensions is about 26 K words or 26 pages in the HP-1000 system. If a larger number of nodal points is required, then the

extended memory (EMA) of the system has to be used for additional storage. The operating system of the HP-1000 computer on which the code is running is the RTE-IVB.

Program KYMA

This is the main part of the code and is used as a driver for the calling sequence to the different subroutines. The definitions of the names of the variables and parameters used in the code are given in the listing of the code in the appendix. The first parameter that needs to be defined is the logical unit (LU) number for the input device, i.e., LU = 1, the input device is the users terminal. The call to INPUT sets the parameters and defines the problem to be solved. Next the user has to specify the time for starting the output of the results (TSTART) and the time to stop this task (TSTOP). If TSTART $\leq$ TIME $\leq$ TSTOP then subroutines XOUT is called. For each time increment a call to QSOLV is required for the solution of the system of equations. This process is continued until the solution reaches the specified maximum time for the simulation.

INPUT

This is the primary parameter definition subroutine. The number of elements for the space discretization, the length of the channel and the water depth are first defined. Next the time increment Δt is defined together with the maximum time for the solution. The call to GEOMT will define the particular channel geometry. Following that step the user has to specify the type of boundary condition to be imposed at the inlet of the channel. At the outlet of the channel the boundary condition is adjusted accordingly by the code.

NOTE: The number of elements should be an even number. In the case of the half cylinder geometry this number should be a multiple of twenty.

GEOMT

This subroutine is used to generate the channel geometry. First the user specifies the type of geometry desired (more types can be added), then whether the channel is of infinite length (open channel) or finite length (closed channel) is considered. After the generation of the specified geometry a warning is given if the number of elements has been increased. This increase is necessary if the spacing of the nodes has been changed to accommodate a particular geometry. The different types of geometries currently generated by the code are shown in Fig. 2.

ELMAT

This subroutine is called by QSOLV and is used to define the element matrices $[M]$ and $[K]$, Eq. (3), and to assemble the element matrices $[A]$ and $[B]$.

QSOLV

This subroutine performs a number of tasks. First the boundary conditions for the specified type are defined. Then the boundary force vector is computed and the global matrices are assembled from the element matrices defined in ELMAT. A sequence of calls to MPART, DCOMP, and SOLVE will partition the global matrices according to the boundary conditions and will solve the system of equations, respectively.

XOUT

This subroutine is called by the main program and it is used for the output of results at selected devices. The user has the choice of output on a line printer, on magnetic tape or at the user's terminal.

XPLOT

This subroutine is used to plot the results at selected time intervals and it is called by XOUT. Results for the elevation, the velocity or the pressure can be plotted individually or in any combination.

TOUT

This subroutine is also called by the main program and it is used for the output of results as a function of time and at specific points with respect to the x-coordinate. Currently the values of the elevation and velocity at $x = 0$ and $x = L = (100.00)$ are obtained.

TPLOT

This subroutine is called by TOUT for plotting results at the user's terminal. Both XPLOT and TPLOT are written by utilizing the capabilities of the HP-2648 graphics terminal.

MPAPT

This subroutine performs the partitioning of the matrix coefficient of the system of equations according to the specified boundary conditions.

DCOMP and SOLVE

These two subroutines perform the decomposition and solution of the matrix systems of equations.

The last three subroutines are part of a users library on the HP-1000 computer system. This reduces the size of the computer code and the compiling time.

4. Program Verification

The required parameters for solving a test case with the developed computer code are specified by the user through subroutine INPUT.

The example which will be used as a demonstration case simulates the propagation of a surface wave in a channel with a half-cylinder placed on the channel bottom. A list of all the input parameters as they appear on the users terminal is given in Appendix A.

The first parameter is the number of elements (NUMEL) to be used for the space discretization, NUMEL = 100. The length of the computational domain is XL = 100.00 m and the water depth is $H_0 = 10.0$ m. The length of each element is equal to 1 m and the time step size is specified as $Dt =$ 0.02. The channel geometry is given in Fig. 3 for the problem under consideration. For this type of geometry NUMEL has been increased to 110. This is a result of the finer mesh where the half-cylinder is positioned. The element length has been reduced to 0.5 at that location.

The physical problem is defined by specifying the type of boundary conditions at $x = 0.0$. In the present case the velocity is specified as

$$V(0,\ t) = V_0 \sin(\omega t)$$

where the amplitude $V_0 = 0.1$ and $\omega = 2\pi/T$. The wave period T is equal to 5.0 which corresponds to a wavelength of 50 or five times the depth H_0.

The rest of the program parameters are needed for the output of results on a printer and/or tape and for the plotting results on the users terminal. Examples of the plotted results are given in Figs. 4 through 6. In Fig. 4 a continuous plot is given for the surface elevation. On the left side of the cylinder one can observe the formation of the standing wave due to the wave reflections from the cylinder. The actual surface elevation as well as the elevation with respect to the size of the channel are given on the same figure. The next two figures are for the elevation and velocity at $x = 0.0$ and $x = 100.0$. The change in elevation due to wave reflections can be seen in Fig. 5. These results have been compared to analytical and experimental ones and the agreement was very satisfactory [2].

The particular problem presented here is only an example of the type of problems that can be solved with the developed code. Additional types of geometries and boundary conditions can be easily handled by the code.

IV. CONCLUDING REMARKS

A finite element model has been implemented for simulating the motion of surface waves in open channels. The computer code which was developed for that purpose has been presented in this report. The description of the code together with an example problem provide the prospective user with sufficient information for implementing and using the code. At this point the code can be used on any machine with a FORTRAN compiler. The only exception is for the two plotting subroutines. These are designed to be used only with the HP-2648 terminals. The code has been optimized to a certain degree but further savings in execution time can be achieved by implementing the vector instruction set (VIS) available on the HP-1000 computer system.

ACKNOWLEDGMENT

The authors acknowledge the support of the Naval Research Laboratory and the Summer Research Apprentice Program for the work reported in this report.

REFERENCES

1. G.A. Keramidas, "A Hydrodynamic Model for Open Channel Flow Problems," Finite Elements in Water Resources, 4th Int'l. Conf., Hannover, Germany, (1982).

2. G.A. Keramidas and S.E. Ramberg, "Numerical Experiments with Reflecting Water Waves," in *Computational Methods and Experimental Measurements*, Washington, D.C., (1982).

3. G.A. Keramidas, "Computational Hydrodynamics: A Model for the Shallow Water Equations," NRL Memorandum Report.

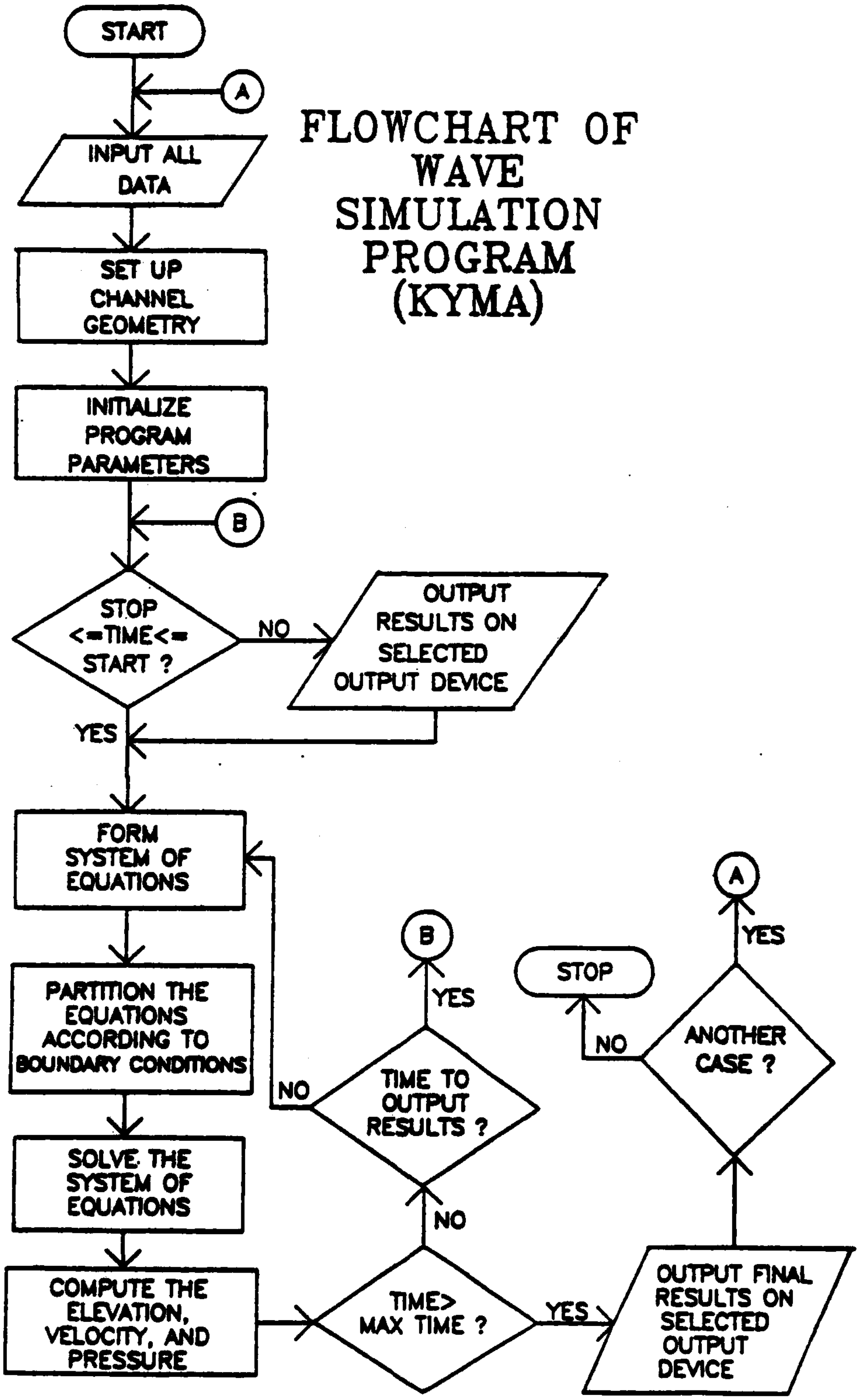


Fig. 1 — Flowchart of computer code

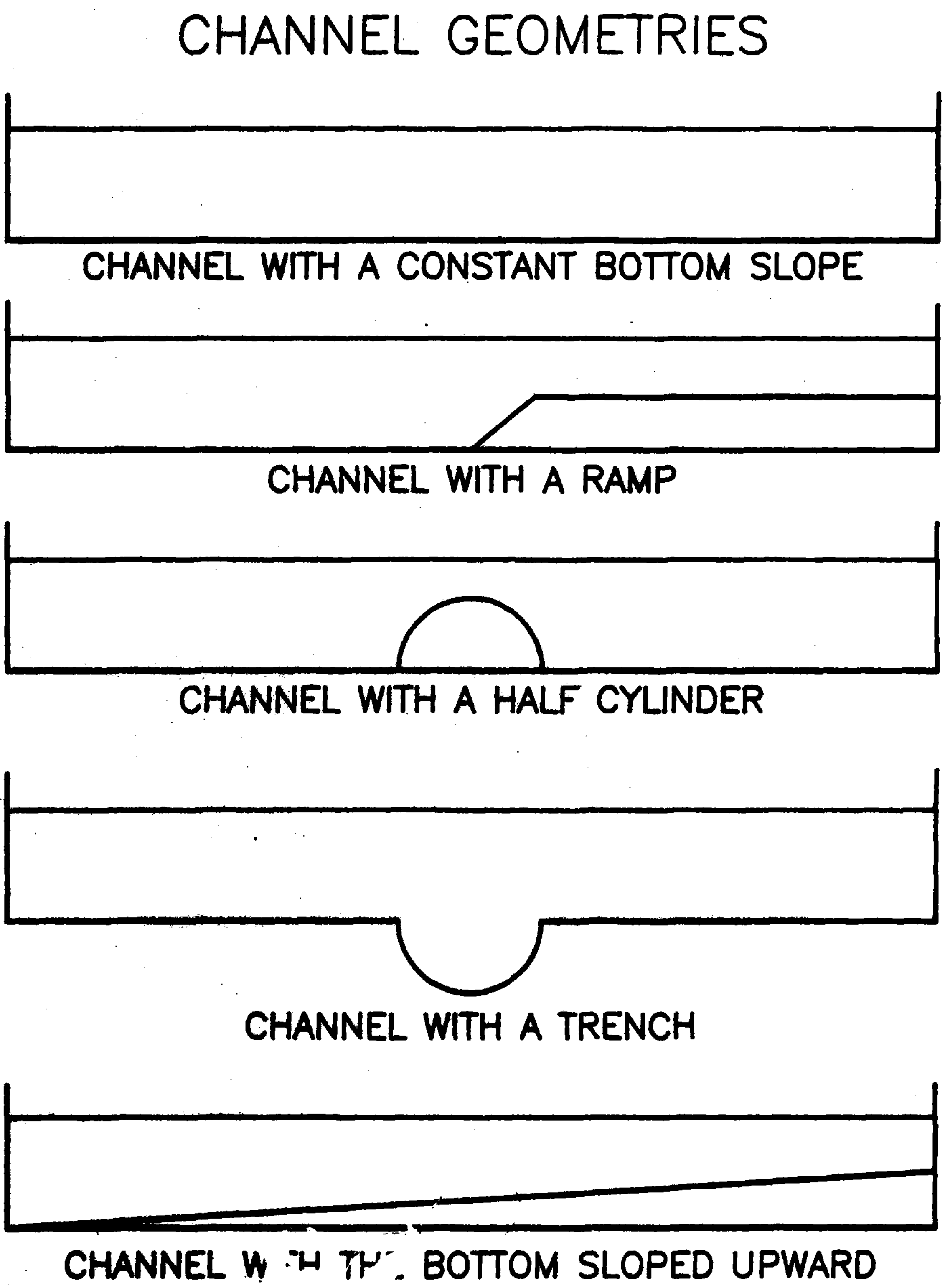


Fig. 2 — Channel geometries presently available in the computer code

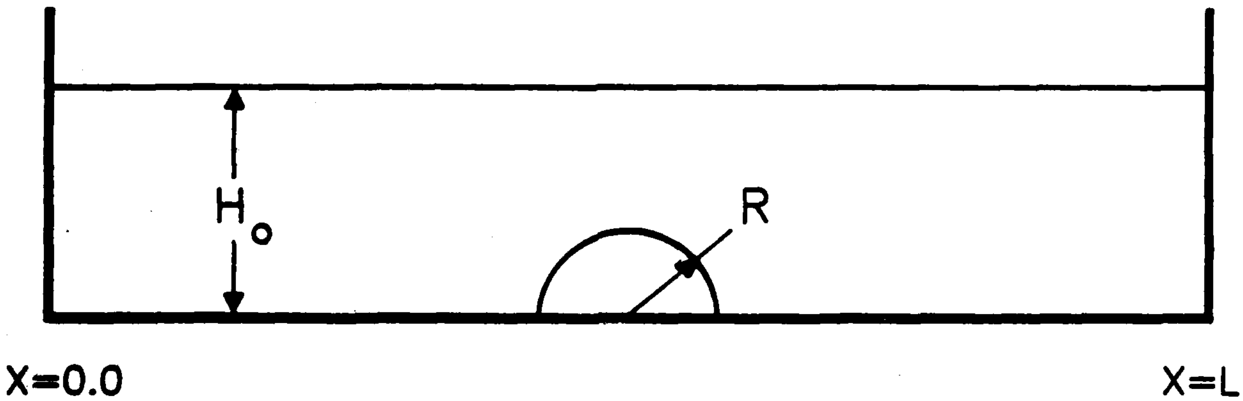


Fig. 3 — Channel geometry for example problem. Half-cylinder with radius $R = 5.0$ m and $H_0 = 10.0$ m.

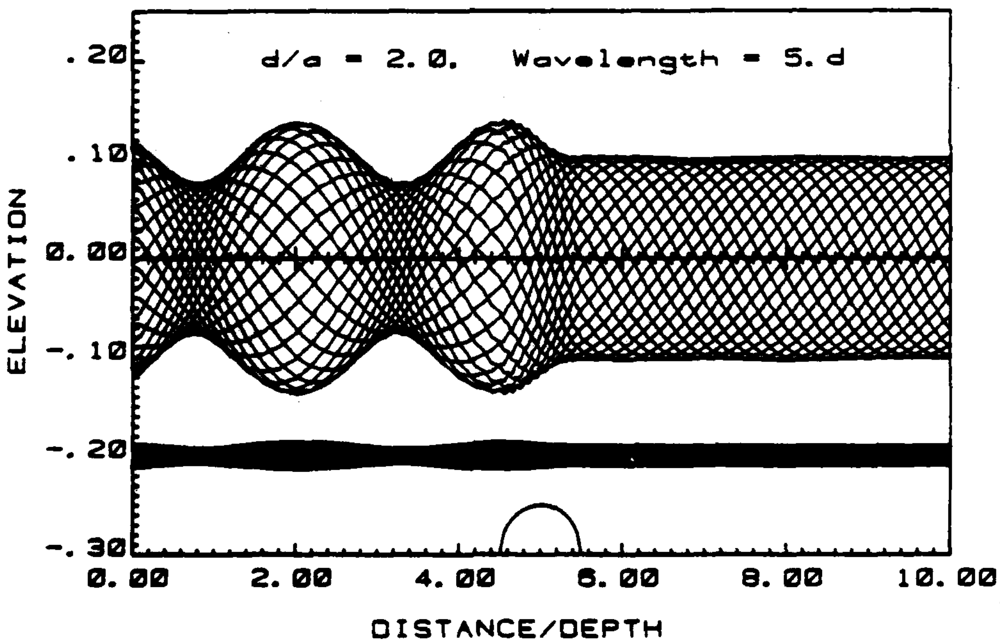


Fig. 4 — Surface elevation along channel during 25 cycles with half-cylinder on the bottom. Nonreflecting boundary at the right.

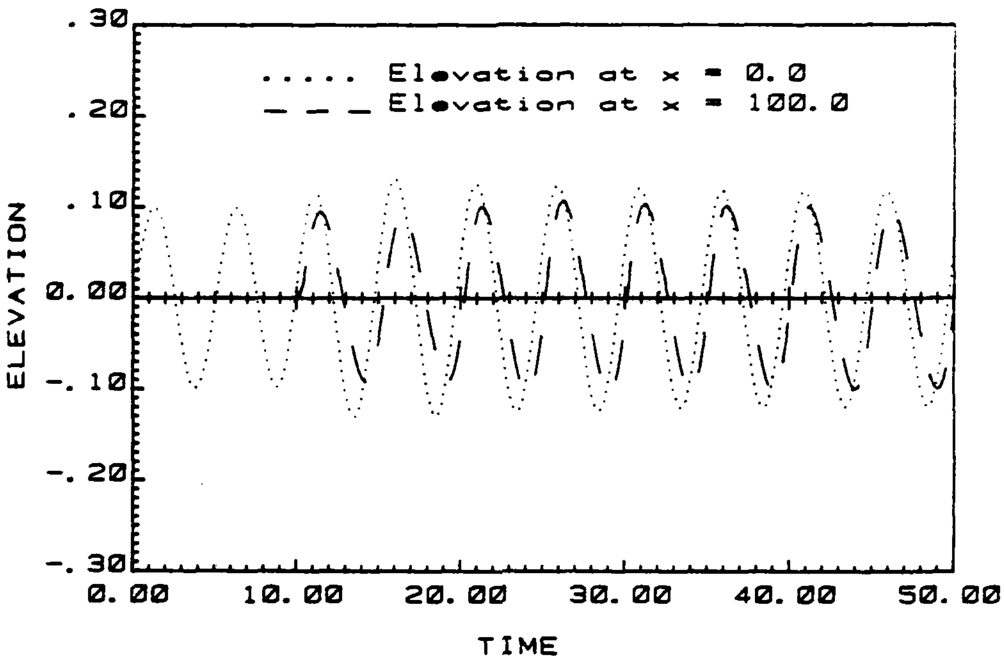


Fig. 5 — Time history of wave elevation at the left and right boundaries for the case shown in Fig. 4

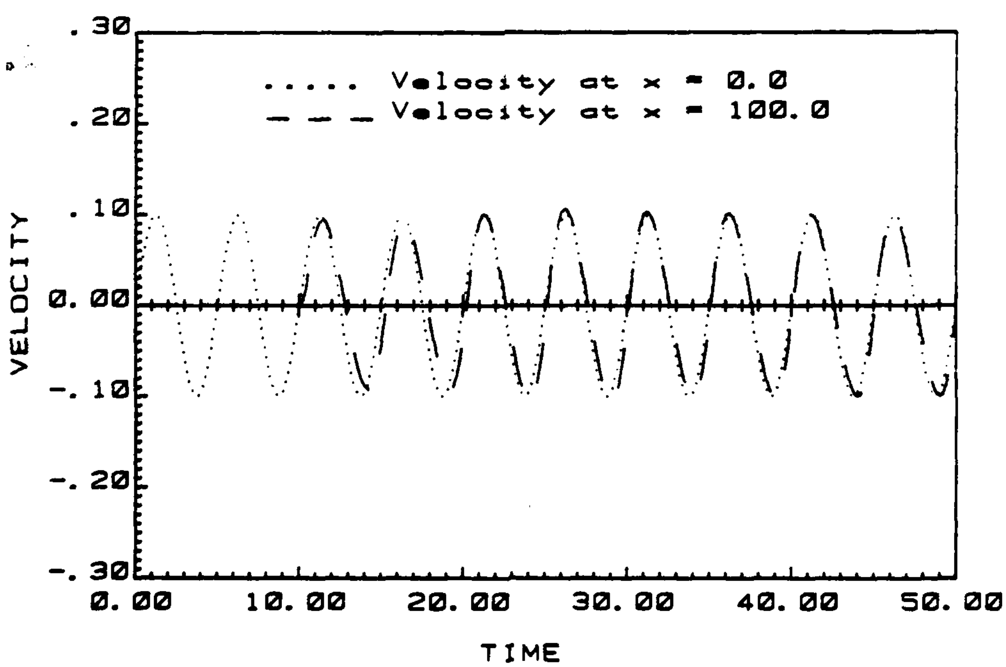


Fig. 6 — Time history of velocity at the left and right boundaries for the case shown in Fig. 4

```

************************************************************
****                                                    ****
****                                                    ****
****             COMPUTATIONAL HYDRODYNAMICS            ****
****                                                    ****
****              SIMULATION OF WATER WAVES             ****
****                        IN  AN                      ****
****                    OPEN  CHANNEL                   ****
****                                                    ****
************************************************************
************************************************************
*************** LU FOR INPUT OF DATA ***********************
**** LUIN # = 1, TERM  IS THE INPUT DEVICE              ****
**** LUIN # = 4, LEFT CASSETE IS THE INPUT DEV.         ****
**** LUIN # = 5, RIGHT CASSETE IS THE INPUT DEV.        ****
**** LUIN # = 8, TAPE DRIVE IS THE INPUT DEV.           ****
**** TYPE THE LU # = 1
************************************************************
**** INPUT NUMBER OF ELEMENTS. IF A HALF CYLINDER       ****
**** IS GOING TO BE USED FOR THE CHANNEL GEOMETRY,      ****
**** THE NUMBER OF ELEMENTS SHOULD BE A MULTIPLE OF     ****
**** 20. NELEM = 100
************************************************************
**** THE LENGTH OF EACH ELEMENT L IS DEFINED AS         ****
****              DX(L) = XL/NEMEL                      ****
**** WHERE XL IS THE LENGTH OF THE CHANNEL              ****
************************************************************
************************************************************
**** INPUT THE CHANNEL LENGTH, (XL =100.0), XL = 100
************************************************************
**** INPUT THE WATER DEPTH H0 = 10
************************************************************
**** INPUT THE TIME STEP SIZE DT = .02
************************************************************
**** INPUT THE MAX TIME FOR THE SIMULATION,TMAX = 1.
************************************************************
**** HOW FREQUENTLY WOULD YOU LIKE THE RESULTS          ****
**** OUTPUTTED? IF IFREQ=10, THEN RESULTS WILL BE       ****
**** OUTPUTTED EVERY 10 TIME STEPS. IFREQ = 10
************************************************************
**** SPECIFY THE TYPE OF CHANNEL GEOMETRY               ****
****         GEOM=0   CONSTANT BOTTOM SLOPE             ****
****         GEOM=1   CHANNEL WITH A RAMP               ****
****         GEOM=2   HALF-CYLINDER ON THE BOTTOM       ****
****         GEOM=3   BOTTOM WITH CYLINDRICAL TRENCH    ****
****         GEOM=4   BOTTOM SLOPED UPWARD              ****
**** GEOM = 2
************************************************************
**** SPECIFY THE TYPE OF CHANNEL                        ****
****         TYPE=1   INFINITE CHANNEL                  ****
****         TYPE=2   FINITE CHANNEL                    ****
**** TYPE = 1
************************************************************
**** SPECIFY THE TYPE OF BOUNDARY CONDITION AT INLET****
****                                                    ****
****    IBC=1,H(1)=CONSTANT                             ****
****    IBC=2,H(1)=SINE WAVE                            ****
```

```
****     IBC=3,V(1)=CONSTANT                                ****
****     IBC=4,V(1)=SINE WAVE                               ****
****     IBC=5,H(1)=CONSTANT,FINITE CHANNEL                 ****
****     IBC=6,H(1)=SINE WAVE,FINITE CHANNEL                ****
**** IBC = 2
*************************************************************
**** IF YOU WANT TO INCLUDE THE BOTTOM FRICTION             ****
**** TERM INTO THE MOMENTUM EQUATION,                       ****
**** TYPE 1 FOR YES, 0 FOR NO : 0
*************************************************************
****SPECIFY THE VALUE OF THE WAVE AMPLITUDE, AMO = .1
*************************************************************
****SPECIFY THE VALUE OF THE WAVE PERIOD. TO = 5
*************************************************************
**** INPUT THE TIME TO START THE OUTPUT OF RESULT           ****
**** TSTAR = 0.
*************************************************************
**** INPUT THE TIME TO STOP THE OUTPUT OF RESULTS           ****
**** TSTOP = 1.
     TIME = 0.000
     TIME =  .200
*************************************************************
**** DO YOU WANT TO OUTPUT THE RESULTS ON THE               ****
**** PRINTER? TYPE 1 FOR YES, 0 FOR NO : 0
*************************************************************
**** DO YOU WANT TO OUTPUT THE RESULTS ON THE TAPE          ****
**** DRIVE? TYPE 1 FOR YES, 0 FOR NO : 0
*************************************************************
****SPECIFY WHICH OF THE FOLLOWING VARIABLES YOU            ****
****WANT PLOTTED :                                          ****
****          0 = NOTHING                                   ****
****          1 = ELEVATION                                 ****
****          2 = VELOCITY                                  ****
****          3 = PRESSURE                                  ****
****          4 = ELEVATION AND VELOCITY                    ****
****          5 = VELOCITY AND PRESSURE                     ****
****          6 = ELEVATION AND PRESSURE                    ****
****          7 = ELEVATION,VELOCITY,AND PRESSURE           ****
**** IPLOT = 0
     TIME =  .400
     TIME =  .600
     TIME =  .800
     TIME = 1.000
          NUMBER OF POINTS, NP =  111
 THE # OF POINTS FOR THE TIME PLOT IS =   5
*************************************************************
**** DO YOU WANT PRINTER OUTPUT OF THE NUMBER OF            ****
**** POINTS,PLOTS,ETC.? TYPE 1 FOR YES, 0 FOR NO : 0
*************************************************************
**** DO YOU WANT TAPE OUTPUT OF ELEVATION AND               ****
**** VELOCITY? TYPE 1 FOR YES, 0 FOR NO : 0
READY TO PLOT GRAPHICS, PRESS RETURN
/
*************************************************************
```

```
FTN7X,L
        PROGRAM KYMA(3,95), UPDATED 9:20:82
C*****
C*************************************************************
C*****                                                   *****
C*****THIS PROGRAM SOLVES THE DEPTH AVERAGED CONTINUITY   *****
C*****AND MOMENTUM EQUATIONS IN ONE DIMENSION. THE        *****
C*****COMPUTATIONAL MODEL IMPLEMENTED IN THIS PROGRAM     *****
C*****IS BASED ON THE FINITE ELEMENT METHOD. THE MODEL    *****
C*****IS DERIVED BY ASSUMING LINEAR APPROXIMATION IN      *****
C*****SPACE AND TIME.                                     *****
C*****               FLUID DYNAMICS BRANCH                *****
C*****             NAVAL RESEARCH LABORATORY              *****
C*****                                                   *****
C*************************************************************
C*****
      COMMON/BLOC1/ T0,T,TMAX,IFREQ,IBC,NUMBC,IB(2)
      COMMON/BLOC2/ Q(222),YQ(222)
      COMMON/BLOC3/ H(120),V(120),DH(120),DV(120),PR(120)
      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
      COMMON/BLOC5/ N,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
      COMMON/BLOC6/ HTN(151),VTN(151),HT1(151),VT1(151),TIM(151)
      DIMENSION GSM(2505),IP(222)
C*****
      WRITE(1,1000)
 1000 FORMAT("************************************************************"
     1    ,/,"****                                                    ****"
     2    ,/,"****                                                    ****"
     3    ,/,"****          COMPUTATIONAL HYDRODYNAMICS               ****"
     4    ,/,"****                                                    ****"
     5    ,/,"****           SIMULATION OF WATER WAVES                ****"
     6    ,/,"****                    IN  AN                          ****"
     7    ,/,"****                 OPEN  CHANNEL                      ****"
     8    ,/,"****                                                    ****"
     9    ,/,"************************************************************")
      GO TO 5
  100 CONTINUE
      WRITE(1,1040)
      WRITE(1,1005)
 1005 FORMAT("************************************************************"
     1    ,/,"**** DO YOU WANT TO RUN ANOTHER CASE?                   ****"
     2    ,/,"**** TYPE 1 FOR YES, 0 FOR NO : _")
      READ(1,*) ICASE
      IF(ICASE.LT.1) GO TO 9999
      WRITE(1,1010)
 1010 FORMAT("************************************************************"
     1    ,/,"**** WOULD YOU LIKE TO KEEP SOME OF THE DATA THE        ****"
     2    ,/,"**** SAME? TYPE 1 FOR YES, 0 FOR NO : _")
      READ(LUIN,*) IQUES
      IF(IQUES.LT.1) ICALL=0
      IF(IQUES.LT.1) GO TO 5
      WRITE(1,1015)
 1015 FORMAT("************************************************************"
     1    ,/,"**** WHICH SET OF DATA WOULD YOU LIKE TO CHANGE?        ****"
     2    ,/,"****  0=MORE THAN ONE SET                               ****"
     3    ,/,"****  1=LU # FOR INPUT OF DATA                          ****"
     4    ,/,"****  2=INPUT DATA(CHANNEL LENGTH,TIME STEP,ETC.)       ****"
     5    ,/,"****  3=CHANNEL GEOMETRY                                ****"
```

```
      6     ,/,"**** WHICH SET? _")
       READ(LUIN,*) ICALL
       IF(ICALL.GT.1) GO TO 6
     5 WRITE(1,1020)
  1020 FORMAT("**************************************************************"
      1     ,/,"**************** LU FOR INPUT OF DATA *****************"
      2     ,/,"**** LUIN # = 1, TERM  IS THE INPUT DEVICE            ****"
      3     ,/,"**** LUIN # = 4, LEFT CASSETE IS THE INPUT DEV.       ****"
      4     ,/,"**** LUIN # = 5, RIGHT CASSETE IS THE INPUT DEV.      ****"
      5     ,/,"**** LUIN # = 8, TAPE DRIVE IS THE INPUT DEV.         ****"
      6     ,/,"**** TYPE THE LU # = _")
        READ(1,*) LUIN
        IF(ICALL.EQ.1) GO TO 7
C*****
      6  CALL INPUT(LUIN,ICALL)
C*****
      7  IF(N.EQ.0) GO TO 9999
C*****
         PI=3.14159
         GE=10.0           !GRAVITATIONAL ACCELERATION
         C0=SQRT(H0*GE)    !WAVE VELOCITY
C*****
         NUMEL=N           !NUMBER OF ELEMENTS
         NPOIN=N+1         !NUMBER OF NODAL POINTS
        NEQ=2*NPOIN        !NUMBER OF EQUATIONS
C*****
C*****INITIALIZE PROGRAM PARAMETERS
C*****
          T=0.0
         DO 10 I=1,NEQ
         IP(I)=0.0         !VECTOR FOR PIVOTING(DCOMP,SOLVE)
          Q(I)=0.0         !VECTOR OF NODAL UNKNOWNS(TIME T)
         YQ(I)=0.0         !VECTOR OF NODAL UNKNOWNS(TIME T-DT)
    10   CONTINUE
         DO 20 I=1,NPOIN
          V(I)=0.0         !VELOCITY VECTOR
          H(I)=0.0         !ELEVATION VECTOR
         RH(I)=HC(I)       !TOTAL WATER DEPTH VECTOR
         PR(I)=0.0         !PRESSURE VECTOR
    20  CONTINUE
C*****
          NBAND=7          !BANDWIDTH OF MATRIX EQUATION
          NHBAN=(NBAND-1)/2
            JGF=NHBAN*NEQ
           JGSM=JGF+NEQ
           JEND=JGSM+NEQ*NBAND
          IF(JEND.GT.2505) GO TO 8888
        ICUNTE=0
            IT=0
           IPT=0
         NPLOT=0
C*****
       WRITE(1,1025)
  1025 FORMAT("**************************************************************"
      1     ,/,"**** INPUT THE TIME TO START THE OUTPUT OF RESULT    ****"
      2     ,/,"**** TSTAR = _")
       READ(LUIN,*) TSTAR
       WRITE(1,1030)
  1030 FORMAT("**************************************************************"
      1     ,/,"**** INPUT THE TIME TO STOP THE OUTPUT OF RESULTS    ****"
```

```
     2    ,/,"**** TSTOP = _")
      READ(LUIN,*) TSTOP
  150 CONTINUE
      WRITE(1,1035) T
      IF(T.LE.TSTAR.OR.T.GE.TSTOP) GO TO 200
C*****
      CALL XOUT(LUIN,IPT)
C*****
C*****STORE H(0,T),H(L,T),V(0,T),V(L,T)
C*****
        IT=IT+1
        HTN(IT)=H(NPOIN)         !H(L,T)
        VTN(IT)=V(NPOIN)         !V(L,T)
        HT1(IT)=H(1)             !H(0,T)
        VT1(IT)=V(1)             !V(0,T)
        TIM(IT)=T
       IF(IT.GE.151) GO TO 46
C*****
  200     T=T+DT
C*****
C*****THE ARRAY GSM(I) IS USED FOR STORING THE NBAND
C*****DIAGONALS OF THE STIFNESS MATRIX
C*****GSM(I), I=JGF+1,JGSM, RIGHT HAND SIDE VECTOR OR
C*****                      SOLUTION VECTOR.
C*****GSM(I), I=JGSM+1,JEND, 1ST,2ND,...,7TH NON-
C*****                      ZERO GIAGONALS.
C*****
       DO 30 I=1,JEND
   30    GSM(I)=0.0
C*****
       CALL QSOLV(GSM(1),JEND,IP,NEQ,NBAND)
C*****
        DO 40 I=1,NEQ
          Q(I)=GSM(I+JGF)
   40  CONTINUE
C*****
C*****COMPUTE THE ELEVATION, VELOCITY & PRESSURE
C*****
       DO 45 I=1,NPOIN
          H(I)=Q(2*I-1)
          V(I)=Q(2*I)
         RH(I)=HC(I)+H(I)
         PR(I)=H(I)+0.5*V(I)**2
   45  CONTINUE
C*****
C*****IF T EXCEEDS TMAX, TERMINATE INTEGRATION..................
C*****
      IF(T.LE.TMAX) GO TO 50
C*****
      GO TO 48
   46 WRITE(1,47)
   47 FORMAT("****************** WARNING **************************"
     1    ,/,"**** THE NUMBER OF POINTS FOR THE TIME PLOTS HAS     ****"
     2    ,/,"**** EXCEEDED THE MAXIMUM NUMBER OF POINTS ALLOWED.  ****"
     3    ,/,"**** INCREASE IFREQ FOR LESS TIME PLOTS IF ANOTHER   ****"
     4    ,/,"**** CASE IS DESIRED.                                ****"
     5    ,/,"*********************************************************")
C*****
   48 CALL TOUT(LUIN,NPOIN,IT,IPT)
C*****
```

```
          GO TO 100
     50  ICUNTE=ICUNTE+1
          DO 55 I=1,NEQ
            YQ(I)=Q(I)
     55  CONTINUE
C*****
C............PRINT RESULTS OR CONTINUE...............
C*****
          IF (ICUNTE.NE. IFREQ) GO TO 200
          ICUNTE=0
C*****
C*****
          GO TO 150
 8888 CONTINUE
 9999  STOP
C*****
 1035  FORMAT (//,5X,6HTIME =,F6.3,//)
 1040  FORMAT("ξ*dD")
          END
C*****
```

```
&KYMA  T=00003 IS ON CR00011 USING 00168 BLKS R=0000
```

```
C***********************************************
C*****
      SUBROUTINE INPUT(LUIN,ICALL), N.C.CHU
C*****
      COMMON/BLOC1/ T0,T,TMAX,IFREQ,IBC,NUMBC,IB(2)
      COMMON/BLOC2/ Q(222),YQ(222)
      COMMON/BLOC3/ H(120),V(120),DH(120),DV(120),PR(120)
      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
      COMMON/BLOC5/ N,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
C*****
      IF(ICALL.LT.1) GO TO 10
      WRITE(1,1000)N
 1000 FORMAT("***********************************************************"
     1    ,/,"**** HIT THE CARRIAGE RETURN FOR ANY VARIABLE THAT    ****"
     2    ,/,"**** YOU WOULD LIKE TO REMAIN THE SAME.                ****"
     3    ,/,"****>>>>>>>>>>>>>>>>>>  WARNING  <<<<<<<<<<<<<<<<<<<<****"
     5    ,/,"**** THE ELEMENT NUMBER HAS BEEN CHANGED TO ",I4," ****")
   10 WRITE(1,1010)
 1010 FORMAT("***********************************************************"
     1    ,/,"**** INPUT NUMBER OF ELEMENTS. IF A HALF CYLINDER     ****"
     2    ,/,"**** IS GOING TO BE USED FOR THE CHANNEL GEOMETRY,    ****"
     3    ,/,"**** THE NUMBER OF ELEMENTS SHOULD BE A MULTIPLE OF ****"
     4    ,/,"**** 20. NELEM = _")
      READ(LUIN,*) N
      IF(ICALL.EQ.3) GO TO 20
      WRITE(1,1015)
 1015 FORMAT("***********************************************************"
     1    ,/,"**** THE LENGTH OF EACH ELEMENT L IS DEFINED AS       ****"
     2    ,/,"****              DX(L) = XL/NEMEL                     ****"
     3    ,/,"**** WHERE XL IS THE LENGTH OF THE CHANNEL            ****"
     4    ,/,"***********************************************************")
      WRITE(1,1020)
 1020 FORMAT("***********************************************************"
     1    ,/,"**** INPUT THE CHANNEL LENGTH, (XL =100.0), XL = _")
      READ(LUIN,*) XL
   20 W0=FLOAT(N)/XL
      NPOIN=N+1
      IF(ICALL.EQ.3) GO TO 30
      WRITE(1,1025)
```

```
 1025 FORMAT("**************************************************************"
     1     ,/,"**** INPUT THE WATER DEPTH H0 = _")
      READ(LUIN,*) H0
      WRITE(1,1030)
 1030 FORMAT("**************************************************************"
     1     ,/,"**** INPUT THE TIME STEP SIZE DT = _")
      READ(LUIN,*)DT
      WRITE(1,1035)
 1035 FORMAT("**************************************************************"
     1     ,/,"**** INPUT THE MAX TIME FOR THE SIMULATION,TMAX = _")
      READ(LUIN,*) TMAX
      WRITE(1,1040)
 1040 FORMAT("**************************************************************"
     1     ,/,"**** HOW FREQUENTLY WOULD YOU LIKE THE RESULTS          ****"
     2     ,/,"**** OUTPUTTED? IF IFREQ=10, THEN RESULTS WILL BE       ****"
     3     ,/,"**** OUTPUTTED EVERY 10 TIME STEPS. IFREQ = _")
      READ(LUIN,*) IFREQ
C*****
C*****SET CHANNEL GEOMETRY
C*****
   30 CALL GEOMT(NPOIN,LUIN,IT,ICALL)
C*****
      IF(ICALL.EQ.3) GO TO 40
      WRITE(1,1045)
 1045 FORMAT("**************************************************************"
     1     ,/,"**** SPECIFY THE TYPE OF BOUNDARY CONDITION AT INLET****"
     2     ,/,"****                                                    ****"
     3     ,/,"****     IBC=1,H(1)=CONSTANT                            ****"
     4     ,/,"****     IBC=2,H(1)=SINE WAVE                           ****"
     5     ,/,"****     IBC=3,V(1)=CONSTANT                            ****"
     6     ,/,"****     IBC=4,V(1)=SINE WAVE                           ****"
     7     ,/,"****     IBC=5,H(1)=CONSTANT,FINITE CHANNEL             ****"
     8     ,/,"****     IBC=6,H(1)=SINE WAVE,FINITE CHANNEL            ****"
     9     ,/,"**** IBC = _")
      READ(LUIN,*) IBC
   40 CONTINUE
      NUMBC=2          ! NUMBER OF BOUNDARY CONDITIONS REQUIRED
      GO TO (50,50,51,51,50,50), IBC
   50 IB(1)=1          ! INDEX FOR BOUNDARY CONDITION AT X=0
      IB(2)=2*(N+1)    ! INDEX FOR BOUNDARY CONDITION AT X=100
      GO TO 57
   51 IB(1)=2
      IB(2)=2*(N+1)-1
   57 IF(ICALL.EQ.3) GO TO 80
      WRITE(1,1050)
 1050 FORMAT("**************************************************************"
     1     ,/,"**** IF YOU WANT TO INCLUDE THE BOTTOM FRICTION         ****"
     2     ,/,"**** TERM INTO THE MOMENTUM EQUATION,                   ****"
     3     ,/,"**** TYPE 1 FOR YES, 0 FOR NO : _")
      READ(LUIN,*) ICZ
      IF(ICZ.LE.0) GO TO 59
      WRITE(1,1055)
 1055 FORMAT("**************************************************************"
     1     ,/,"****SPECIFY THE VALUE OF THE CHEZY COEFFICIENT. CZ = _")
      READ(LUIN,*) CZ
   59 WRITE(1,1060)
 1060 FORMAT("**************************************************************"
     1     ,/,"****SPECIFY THE VALUE OF THE WAVE AMPLITUDE. AM0 = _")
      READ(LUIN,*) AM0
      GO TO (80,70,80,70,80,70),IBC
   70 WRITE(1,1065)
```

```
 1065 FORMAT("*************************************************************"
     1     ,/,"****SPECIFY THE VALUE OF THE WAVE PERIOD. T0 = _")
      READ(LUIN,*) T0
   80 RETURN
      END
C*****
```

&KYMA T=00003 IS ON CR00011 USING 00168 BLKS R=0000

```
C*********************************************
C*****
      BLOCK DATA
      COMMON/BLOC1/ T0,T,TMAX,IFREQ,IBC,NUMBC,IB(2)
      COMMON/BLOC2/ Q(222),YQ(222)
      COMMON/BLOC3/ H(120),V(120),DH(120),DV(120),PP(120)
      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
      COMMON/BLOC5/ N,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
      COMMON/BLOC6/ HTN(151),VTN(151),HT1(151),VT1(151),TIM(151)
      END
C*****
```

&KYMA T=00003 IS ON CR00011 USING 00168 BLKS R=0000

```
C**********************************************
C*****
      SUBROUTINE GEOMT(NPOIN,LUIN,IT,ICALL), N.C.CHU, G.A.KERAMIDAS
C*****
      COMMON/BLOC3/ H(120),V(120),DH(120),DV(120),PR(120)
      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
      COMMON/BLOC5/ N,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
C*****
      IF(ICALL.EQ.2) GO TO 5
      WRITE(1,1000)
 1000 FORMAT("*************************************************************"
     1     ,/,"**** SPECIFY THE TYPE OF CHANNEL GEOMETRY                  ****"
     2     ,/,"****           GEOM=0   CONSTANT BOTTOM SLOPE             ****"
     3     ,/,"****           GEOM=1   CHANNEL WITH A RAMP               ****"
     4     ,/,"****           GEOM=2   HALF-CYLINDER ON THE BOTTOM       ****"
     5     ,/,"****           GEOM=3   BOTTOM WITH CYLINDRICAL TRENCH    ****"
     6     ,/,"****           GEOM=4   BOTTOM SLOPED UPWARD              ****"
     7     ,/,"**** GEOM = _")
      READ(LUIN,*) IGEOM
      WRITE(1,1010)
 1010 FORMAT("*************************************************************"
     1     ,/,"**** SPECIFY THE TYPE OF CHANNEL                           ****"
     2     ,/,"****           TYPE=1   INFINITE CHANNEL                  ****"
     3     ,/,"****           TYPE=2   FINITE CHANNEL                    ****"
     4     ,/,"**** TYPE = _")
      READ(LUIN,*) ITYPE
      IF(ITYPE.EQ.1) GO TO 5
      WRITE(1,1015)
 1015 FORMAT("*************************************************************"
     1     ,/,"**** INPUT THE DEPTH OF THE CHANNEL AT THE CLOSED         ****"
     2     ,/,"**** END. HN0 = _")
      READ(LUIN,*) HN0
      GO TO 6
    5 CONTINUE
      IF(IGEOM.NE.4) GO TO 6
      WRITE(1,1020)
```

```
 1020 FORMAT("*************************************************************"
     1     ,/,"**** INPUT THE DEPTH OF THE WATER AT THE END OF      ****"
     2     ,/,"**** THE CHANNEL  DW0 = _")
      READ(LUIN,*) DW0
      HN0=H0-DW0
    6 DX0=1.0/W0                     ! CONSTANT ELEMENT LENGTH = XL/N
      DO 7 I=1,NPOIN
        DX(I)=DX0                    ! ELEMENT LENGTH VECTOR
        HC(I)=H0                     ! WATER DEPTH IN CHANNEL
        RL(I)=DX(I)*FLOAT(I-1)       ! NODAL POINTS COORDINATES
    7 CONTINUE
      IF(ITYPE.EQ.1) GO TO 9
        HC(NPOIN)=HN0
    9 IF(IGEOM.LE.0) GO TO 50
      GO TO (11,21,31,39), IGEOM
C*****
C*****GEOMETRY FOR CHANNEL WITH A RAMP
C*****
   11 CONTINUE
      HN0=5
       NEH=N/2+1
       NEC=N/10
       NED=NEH+NEC
       DO 15 I=NEH,NED
       DX(I)=DX0/2.0
   15  HC(I)=H0-FLOAT(I-NEH)*DX(I)
       NMAX=NPOIN+NEC
       DO 20 I=NED,NMAX
       DX(I)=DX0
   20  HC(I)=H0-HN0
       GO TO 43
C*****
C*****GEOMETRY FOR CHANNEL WITH A HALF-CYLINDER ON THE BOTTOM
C*****
   21     NEC=N/10
          NC0=(N-NEC)/2
          NC1=NC0+1
          NC2=NC0+2*NEC
          DR=XL/10.0
          RR=DR/2.0
C*****
      DO 25 I=1,NC0
            DX(I)=DX0
        DX(I+NC2)=DX0
   25 CONTINUE
C*****
      DO 27 I=1,NPOIN+NEC
        HC(I)=H0
   27 CONTINUE
C*****
      DO 30 I=NC1,NC2
           I1=I-(NEC/2+1)
        DX(I)=DX0/2.0
           XR=-XL/4.0+DX(I)*FLOAT(I1)
        HC(I)=H0-SQRT(RR**2-XR**2)
   30 CONTINUE
C*****
      GO TO 43
C*****
C*****GEOMETRY FOR A CHANNEL WITH A CYLINDRICAL TRENCH ON THE BOTTON
C*****
```

```
   31     NEC=N/10
          NC0=(N-NEC)/2
          NC1=NC0+1
          NC2=NC0+2*NEC
          DR=XL/10.0
          RR=DR/2.0
C*****
      DO 36 I=1,NC0
           DX(I)=DX0
        DX(I+NC2)=DX0
   36 CONTINUE
C*****
      DO 37 I=1,NPOIN+NEC
        HC(I)=H0
   37 CONTINUE
C*****
      DO 38 I=NC1,NC2
           I1=I-(NEC/2+1)
        DX(I)=DX0/2.0
           XR=-XL/4.0+DX(I)*FLOAT(I1)
       HC(I)=H0+SQRT(RR**2-XR**2)
   38 CONTINUE
      GO TO 43
C*****
C*****GEOMETRY FOR A CHANNEL WITH THE BOTTOM SLOPED UPWARD
C*****
   39     SLOPE=HN0/XL
      DO 41 I=1,NPOIN
          HC(I)=H0-DX(I)*SLOPE*FLOAT(I-1)
   41 CONTINUE
C*****
   43      N=N+NEC        ! THE NUMBER OF ELEMENTS HAS BEEN INCREASED
     C                    ! BY NEC FOR THE SPECIFIED GEOMETRY
      DO 45 I=1,N
        RL(I+1)=RL(I)+DX(I)
   45 CONTINUE
C*****
   50 CONTINUE
      RETURN
      END
C*****

&KYMA  T=00003 IS ON CR00011 USING 00168 BLKS R=0000

C**********************************************
C*****
      SUBROUTINE ELMAT(L,ESM1,ESM2), G.A.KERAMIDAS
C*****
C*****SUBROUTINE ELMAT EVALUATES THE ELEMENT MATRICES
C*****
      COMMON/BLOC1/ T0,T,TMAX,IFREQ,IBC,NUMBC,IB(2)
      COMMON/BLOC3/ H(120),V(120),DH(120),DV(120),PR(120)
      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
      COMMON/BLOC5/ N,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
      DIMENSION A(4,4),B(4,4),ESM1(4,4),ESM2(4,4)
C*****
C*****MASS MATRIX, CORRESPONDS TO MATRIX [M] OF EQ. (3)
C*****
         A(1,1)=2./DT
         A(1,2)=0.0
         A(1,3)=1./DT
         A(1,4)=0.0
```

```
      A(2,1)=0.0
      A(2,2)=2./DT
      A(2,3)=0.0
      A(2,4)=1./DT
      A(3,1)=1./DT
      A(3,2)=0.0
      A(3,3)=2./DT
      A(3,4)=0.0
      A(4,1)=0.0
      A(4,2)=1./DT
      A(4,3)=0.0
      A(4,4)=2./DT
C*****
C*****CONVECTIVE MATRIX, CORRESPONDS TO MATRIX [K] OF EQ. (3)
C*****
          C0=SQRT(GE*RH(L))
         DV0=V(L)-V(L+1)
        DIF0=SQRT(2.*DX(L)/(C0*DT))
        DIF1=1./DIF0
        DIF2=DIF0**2
        DIFL=(DIF2*ABS(DV0)+C0*DIF1)/DX(L)
C*****
      HL0=0.5*(HC(L)+HC(L+1))
      HL1=(2.*HC(L)+HC(L+1))/DX(L)
      HL2=(HC(L)+2.*HC(L+1))/DX(L)
      DHDX=(H(L+1)-H(L))/DX(L)
      DVDX=(V(L+1)-V(L))/DX(L)
         DHC=(HC(L+1)-HC(L))/DX(L)
        AVEL=0.5*(V(L)+V(L+1))
        ADEP=0.5*(RH(L)+RH(L+1))
        DISP=0.0
        IF(IC2.GT.0) DISP=GE/(ADEP*CZ**2)
      DIS11=DISP*(V(L)+AVEL)
      DIS22=DISP*(V(L+1)+AVEL)
      DIS12=DISP*AVEL
      DIS21=DISP*AVEL
C*****
      B(2,2)=2.*DVDX+DIFL+DIS11
      B(2,1)=3.*GE/DX(L)
      B(2,4)=DVDX-DIFL+DIS12
      B(2,3)=3.*GE/DX(L)
      B(1,2)=2.*(DHDX+DHC)-HL1
      B(1,1)=2.*DVDX
      B(1,4)=DHDX+DHC+HL1
      B(1,3)=DVDX
      B(4,2)=DVDX-DIFL+DIS21
      B(4,1)=-3.*GE/DX(L)
      B(4,4)=2.*DVDX+DIFL+DIS22
      B(4,3)=-3.*GE/DX(L)
      B(3,2)=DHDX+DHC-HL2
      B(3,1)=DVDX
      B(3,4)=2.*(DHDX+DHC)+HL2
      B(3,3)=2.*DVDX
C*****
      DO 10 I=1,4
      DO 10 J=1,4
C*****
C*****ELEMENT STIFFNESS MATRICES,
C*****
        ESM1(I,J)=A(I,J)+0.5*B(I,J)     !MATRIX [A] OF EQ.(4)
        ESM2(I,J)=A(I,J)-0.5*B(I,J)     !MATRIX [B] OF EQ.(4)
   10 CONTINUE
C*****
```

```
      RETURN
      END
C*****

&KYMA  T=00003 IS ON CR00011 USING 00168 BLKS R=0000

C****************************************
C*****
      SUBROUTINE QSOLV(GSM,JEND,IP,NEQ,NBAND), G.A.KERAMIDAS
C*****
C*****SUBROUTINE QSOLV FORMS THE SYSTEM OF ALGEBRAIC EQUATIONS
C*****AND CALLS SUBROUTINE MPART TO PARTITION THE EQUATIONS ACCORDING
C*****TO THE BOUNDARY CONDITIONS. IT ALSO CALLS SUBROUTINES DCOMP
C*****AND SOLVE FOR THE SOLUTION OF THE SYSTEM OF EQUATIONS.
C*****
      COMMON/BLOC1/ T0,T,TMAX,IFREQ,IBC,NUMBC,IB(2)
      COMMON/BLOC2/ Q(222),YQ(222)
      COMMON/BLOC3/ H(120),V(120),DH(120),DV(120),PR(120)
      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
      COMMON/BLOC5/ N,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
      DIMENSION GSM(JEND),IP(NEQ)
      DIMENSION ESM1(4,4),ESM2(4,4),EF(4),NS(4),BV(2)
C*****
            NUMEL=N
            NPOIN=N+1
          NHBAN=(NBAND-1)/2      ! HALF BANDWIDTH
           NORD=2                ! DEGREES OF FREEDOM PER NODE
          NDOFR=2*NORD           ! DEGREES OF FREEDOM PER ELEMENT
           JGF=NHBAN*NEQ
          JGSM=JGF+NEQ
            PI=3.14159
            C0=SQRT(H0*GE)
C*****
      DO 10 I=1,4
      DO 10 J=1,4
        ESM1(I,J)=0.0
   10   ESM2(I,J)=0.0
        DO 15 I=1,NDOFR
   15   NS(I)=I-NORD
C*****
      DO 20 I=1,JEND
   20   GSM(I)=0.0
C*****
C*****BOUNDARY CONDITIONS
C*****
      GO TO (21,22,23,24,25,26), IBC
   21       H(1)=AM0
            Q(1)=H(1)
          V(NPOIN)=H(NPOIN)*SQRT(GE*HC(NPOIN))/HC(NPOIN)
         Q(NEQ)=V(NPOIN)
      GO TO 30
   22       H(1)=AM0*SIN(2.*PI*T/T0)
            Q(1)=H(1)
          V(NPOIN)=H(NPOIN)*SQRT(GE*HC(NPOIN))/HC(NPOIN)
         Q(NEQ)=V(NPOIN)
      GO TO 30
   23       V(1)=AM0
            Q(2)=V(1)
          H(NPOIN)=V(NPOIN)*HC(NPOIN)/SQRT(GE*HC(NPOIN))
        Q(NEQ-1)=H(NPOIN)
      GO TO 30
   24       V(1)=AM0*SIN(2.*PI*T/T0)
            Q(2)=V(1)
          H(NPOIN)=V(NPOIN)*HC(NPOIN)/SQRT(GE*HC(NPOIN))
```

```
            Q(NEQ-1)=H(NPOIN)
      GO TO 30
   25          H(1)=AM0
               Q(1)=H(1)
             V(NPOIN)=0.0
            Q(NEQ)=0.0
      GO TO 30
   26          H(1)=AM0*SIN(2.*PI*T/T0)
               Q(1)=H(1)
             V(NPOIN)=0.0
            Q(NEQ)=0.0
   30 CONTINUE
             BV(1)=Q(IB(1))
             BV(2)=Q(IB(2))
C*****
C*****GENERATION OF THE BOUNDARY FORCE VECTOR
C*****
           DO 45 LL=1,NUMEL
             EF(1)=0.00
             EF(2)= 6.*GE*H(LL)/DX(LL)
             EF(3)=0.0
             EF(4)=-6.*GE*H(LL+1)/DX(LL)
C*****
             NS(1)=NS(1)+NORD
             NS(2)=NS(2)+NORD
             NS(3)=NS(3)+NORD
             NS(4)=NS(4)+NORD
           DO 40 I=1,NDOFR
                II=NS(I)
                I2=JGF+II
   40      GSM(I2)=GSM(I2)+EF(I)      ! BOUNDARY FORCE VECTOR
   45      CONTINUE
C*****
           DO 35 I=1,NDOFR
   35      NS(I)=I-NORD
C*****
C*****GENERATION OF THE SYSTEM MATRICES
C*****
           DO 80 KK=1,NUMEL
           CALL ELMAT(KK,ESM1,ESM2)
             NS(1)=NS(1)+NORD
             NS(2)=NS(2)+NORD
             NS(3)=NS(3)+NORD
             NS(4)=NS(4)+NORD
           DO 80 I=1,NDOFR
                II=NS(I)
                J2=JGF+II
           DO 75 J=1,NDOFR
                JJ=NS(J)
           GSM(J2)=GSM(J2)+ESM2(I,J)*YQ(JJ)    ! RIGHT HAND SIDE VECTOR
                JJ=JJ-II+NHBAN
                J5=JGSM+JJ*NEQ+II
           GSM(J5)=GSM(J5)+ESM1(I,J)     ! MATRIX COEFFICIENT OF SYSTEM
   75      CONTINUE
   80      CONTINUE
C*****
           CALL MPART(GSM(JGSM+1),GSM(JGF+1),NEQ,BV,IB,NBAND,NUMBC)
           CALL DCOMP(GSM(1),IP,GSM(JGSM+1),NEQ,NEQ,NHBAN,NBAND)
           CALL SOLVE(GSM(1),IP,GSM(JGF+1),GSM(JGSM+1),NEQ,NEQ,NHBAN,NBAND)
C*****
      RETURN
      END
C*****
```

```
&KYMA  T=00003 IS ON CR00011 USING 00168 BLKS R=0000

C*****************************************
C*****
      SUBROUTINE XPLOT(R,T,N,IPT,IPLOT,ICPL), N.C.CHU, G.A.KERAMIDAS
C*****
C*****SUBROUTINE XPLOT PLOTS THE SOLUTION ACCORDING TO THE X-COORDINATE
C*****
      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
      COMMON/BLOC5/ NP,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
      DIMENSION R(N),H(200),X(120),Y(200)
C*****
      DO 10 I=1,N
      X(I)=RL(I)/10.0
      Y(I)=(H0-HC(I))/100.0-0.30
      H(I)=R(I)/10.0-(0.3-H0/100.)
   10 CONTINUE
C*****
      IF(IPLOT.GT.3) GO TO 979
      IF(ICPL.GT.0) GO TO 47
C*****PLOT GRAPHICS OUTPUT
  979 WRITE (1,980)
  980 FORMAT("READY TO PLOT GRAPHICS, PRESS RETURN")
      READ(1,*) IIC
      IF(IGEOM.NE.3) GO TO 15
  989 FORMAT("ξ*dFξ*dAξ*a0.110.m1.0p0.2q-.3n.25o.05r.01s1vC")
      WRITE(1,981)
  981 FORMAT("ξ*dFξ*dAξ*a0.110.m1.0p0.2q-.35n.25o.05r.01s1vC")
       GO TO 20
   15 WRITE(1,989)
   20 WRITE(1,983)
  983 FORMAT("ξ*a2h1i2j1K")
      WRITE(1,982)
  982 FORMAT("ξ*d300,0okSξ*m2m1Nξ*1X-AXISξ*dT")
       IF(IPT.EQ.0) GO TO (30,40,45,30,40,30,30), IPLOT
       IF(IPT.EQ.1) GO TO (47,47,47,40,45,45,40), IPLOT
       IF(IPT.EQ.2) GO TO (47,47,47,47,47,47,45), IPLOT
   30 WRITE(1,903)
      GO TO 47
   40 WRITE(1,913)
      GO TO 47
   45 WRITE(1,923)
  903 FORMAT("ξ*d30,160okSξ*m2m2Nξ*1ELEVATION  ξ*dT")
  913 FORMAT("ξ*d30,160okSξ*m2m2Nξ*1VELOCITY   ξ*dT")
  923 FORMAT("ξ*d30,160okSξ*m2m2Nξ*1PRESSURE   ξ*dT")
   47 WRITE(1,984)
  984 FORMAT("ξ*aA")
      WRITE(1,987)
  987 FORMAT("ξ*dF")
C*****PLOT COMPUTED SOLUTION AT TIME T
      WRITE(1,985) (X(I),R(I),I=1,N)
      WRITE(1,985) (X(N-I+1),H(N-I+1),I=1,N)
      IF(IPT.GT.1) GO TO 48
      WRITE(1,985) (X(I),Y(I),I=1,N)
  985 FORMAT(2F7.4)
   48 WRITE(1,988)
  988 FORMAT("ξ*dE")
C*****
      WRITE(1,986)
  986 FORMAT("ξ*aBξ*dTξ*dE")
```

```
      WRITE(1,997)
      WRITE(1,998)  T
  997 FORMAT("ξ*d140,310okSξ*m2m1Nξ*1ξMξ*dTξ*m1m1N")
  998 FORMAT("ξ*d140,310okSξ*m2m1Nξ*1 T =",F5.3,"ξ*dTξ*m1m1N")
   50  RETURN
       END
C*****

&KYMA  T=00003 IS ON CR00011 USING 00168 BLKS R=0000

C*****************************************
C*****
      SUBROUTINE TPLOT(R,TIM,N,IPT), G.A.KERAMIDAS
C*****
C*****SUBROUTINE TPLOT PLOTS THE SOLUTION ACCORDING TO THE TIME COORDINATE
C*****
      COMMON/BLOC1/ T0,T,TMAX,IFREQ,IBC,NUMBC,IB(2)
      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
      COMMON/BLOC5/ M,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
      DIMENSION R(N),TIM(N)
C*****
C*****
C. .  PLOT GRAPHICS OUTPUT . . . . . . . . .
      WRITE (1,980)
  980 FORMAT("READY TO PLOT GRAPHICS, PRESS RETURN")
      READ(1,*) II
      WRITE(1,981)
  981 FORMAT("ξ*dFξ*dAξ*a50.0m5,0p1qC")
      WRITE(1,982)
  982 FORMAT("ξ*d300,0okSξ*m2m1Nξ*1T-AXISξ*dT")
      IF(IPT.EQ.0) WRITE(1,903)
      IF(IPT.EQ.1) WRITE(1,913)
      IF(IPT.EQ.2) WRITE(1,923)
  903 FORMAT("ξ*d30,160okSξ*m2m2Nξ*1ELEVATION  ξ*dT")
  913 FORMAT("ξ*d30,160okSξ*m2m2Nξ*1VELOCITY   ξ*dT")
  923 FORMAT("ξ*d30,160okSξ*m2m2Nξ*1ELEVATION  ξ*dT")
      WRITE(1,984)
  984 FORMAT("ξ*aA")
C. . .PLOT COMPUTED SOLUTION AT TIME T1 . . . . . . . . .
      WRITE(1,985) (TIM(I),R(I),I=1,N)
  985 FORMAT(2F7.4)
      WRITE(1,986)
  986 FORMAT("ξ*aBξ*dTξ*dE")
      WRITE(1,998)  T
  998 FORMAT("ξ*d140,310okSξ*m2m1Nξ*1 T =",F4.2,"ξ*dTξ*m1m1N")
      RETURN
      END
C*****

&KYMA  T=00003 IS ON CR00011 USING 00168 BLKS R=0000

C*********************************************
C*****
      SUBROUTINE XOUT(LUIN,IPT), N.C.CHU
C*****
      COMMON/BLOC1/ T0,T,TMAX,IFREQ,IBC,NUMBC,IB(2)
      COMMON/BLOC3/ H(120),V(120),DH(120),DV(120),PP(120)
      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
      COMMON/BLOC5/ N,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
      COMMON/BLOC6/ HTN(151),VTN(151),HT1(151),VT1(151),TIM(151)
C*****
```

```
      NPOIN=N+1
      IF(IPT.GT.0) GO TO 20
      ICPL=0
      ICPL0=0
      WRITE(1,1000)
 1000 FORMAT("**********************************************************"
     1      ,/,"**** DO YOU WANT TO OUTPUT THE RESULTS ON THE         ****"
     2      ,/,"**** PRINTER? TYPE 1 FOR YES, 0 FOR NO : _")
      READ(LUIN,*) IPRIN
      IF(IPRIN.LT.1) GO TO 30
      WRITE(6,1035)
      WRITE(6,1005)
 1005 FORMAT(40X,"**********************************************************
     1**",/,40X,"****                                                   ***
     2* ",/,40X,"****                                                   ***
     3* ",/,40X,"****           COMPUTATIONAL HYDRODYNAMICS             ***
     4* ",/,40X,"****                                                   ***
     5* ",/,40X,"****            SIMULATION OF WATER WAVES              ***
     6* ",/,40X,"****                     IN  AN                        ***
     7* ",/,40X,"****                  OPEN  CHANNEL                    ***
     8* ",/,40X,"****                                                   ***
     9* ",/,40X,"**********************************************************
     1* ")
      C0=SQRT(H0*GE)
      CR0=C0*DT*W0
      WRITE(6,1060)
      WRITE(6,1010) CR0
 1010 FORMAT(40X,"**********************************************************
     1**",/,40X,"****                                                   ***
     2* ",/,40X,"****          THE COURANT NUMBER IS ",F6.3,11X,"****"
     3  ,/, 40X,"****                                                   ***
     4* ",/,40X,"**********************************************************
     5* ")
      WRITE(6,1060)
      WRITE(6,1015) N,XL,DX(1),DT,AM0,IBC
 1015 FORMAT(40X,"**********************************************************
     1**",/,40X,"****                                                   ***
     2* ",/,40X,"****           NUMBER OF ELEMENTS IS ",I6  ,10X,"****"
     3    ,/,40X,"****           LENGTH OF CHANNEL IS ",F6.3,11X,"****"
     4    ,/,40X,"****           LENGTH OF ELEMENT IS ",F6.3,11X,"****"
     5    ,/,40X,"****           TIME STEP SIZE IS ",F6.3,14X,"****"
     6    ,/,40X,"****           AMPLITUDE OF INCOMING WAVE IS ",F6.3,2X,"
     7****",/,40X,"****            BOUDARY CONDITION AT X=0.0 IS IBC=",I2,
     92X,"****",/,40X,"****                 IBC=1,H(1)=CONSTANT"18X,"****"
     9    ,/,40X,"****            IBC=2,H(1)=SINE WAVE                   ***
     1* ",/,40X,"****            IBC=3,V(1)=CONSTANT                    ***
     2* ",/,40X,"****            IBC=4,V(1)=SINE WAVE                   ***
     3* ",/,40X,"****            IBC=5,H(1)=CONSTANT,FINITE CHANNEL     ***
     4* ",/,40X,"****            IBC=6,H(1)=SINE WAVE,INFINITE CHANNEL***
     5* ",/,40X,"****                                                   ***
     6* ",/,40X, "*********************************************************
     7**")
   20 IF(IPRIN.LT.1) GO TO 45
      WRITE(6,1050) T
      WRITE(6,1045)
      WRITE(6,1055) (I,H(I),V(I),PL(I),PH(I),HC(I),PP(I),I=1,NPOIN)
C*****
      IF(IPT.GT.0) GO TO 45
   30 WRITE(1,1020)
 1020 FORMAT("**********************************************************"
     1      ,/,"**** DO YOU WANT TO OUTPUT THE RESULTS ON THE TAPE   ****"
     2      ,/,"**** DRIVE? TYPE 1 FOR YES, 0 FOR NO : _")
      READ(LUIN,*) ITAPE
```

```
   45 IF(ITAPE.LT.1) GO TO 50
      WRITE(8,1040) (H(I),V(I),HC(I),PR(I),RL(I),T,I=1,NPOIN)
   50 IF(IPT.GT.0) GO TO 57
      WRITE(1,1025)
 1025 FORMAT("**************************************************************"
     1     ,/,"****SPECIFY WHICH OF THE FOLLOWING VARIABLES YOU         ****"
     2     ,/,"****WANT PLOTTED :                                        ****"
     3     ,/,"****       0 = NOTHING                                    ****"
     4     ,/,"****       1 = ELEVATION                                  ****"
     5     ,/,"****       2 = VELOCITY                                   ****"
     6     ,/,"****       3 = PRESSURE                                   ****"
     7     ,/,"****       4 = ELEVATION AND VELOCITY                     ****"
     8     ,/,"****       5 = VELOCITY AND PRESSURE                      ****"
     9     ,/,"****       6 = ELEVATION AND PRESSURE                     ****"
     1     ,/,"****       7 = ELEVATION,VELOCITY,AND PRESSURE            ****"
     2     ,/,"**** IPLOT = _")
      READ(LUIN,*) IPLOT
   57 IPT=0
      IF(IPLOT.GT.3) GO TO 58
      IF(ICPL.LE.0.OR.ICPL0.EQ.1) GO TO 58
      WRITE(1,1065)
      WRITE(1,1030)
 1030 FORMAT("**************************************************************"
     1     ,/,"**** DO YOU WANT A CONTINUOUS PLOT?                       ****"
     2     ,/,"**** TYPE 1 FOR YES, 0 FOR NO : _")
      READ(LUIN,*) ICPL
      ICPL0=1
   58 IF(IPLOT.LE.0) GO TO 999
      GO TO (60,65,70,60,65,60,60), IPLOT
   60 CALL XPLOT(H,T,NPOIN,IPT,IPLOT,ICPL)
       IPT=IPT+1
      IF(IPLOT.EQ.1) GO TO 999
      IF(IPLOT.EQ.6) GO TO 70
   65 CALL XPLOT(V,T,NPOIN,IPT,IPLOT,ICPL)
       IPT=IPT+1
      IF(IPLOT.EQ.4) GO TO 999
      IF(IPLOT.EQ.2) GO TO 999
   70 CALL XPLOT(PR,T,NPOIN,IPT.IPLOT.ICPL)
      IPT=IPT+1
  999 IF(ICPL0.EQ.1) GO TO 100
      ICPL=1
C*****
 1035 FORMAT("1")
 1040 FORMAT(5X,6F12.6)
 1045 FORMAT(////,   "   NODAL POINT    H-ELEVATION      V-VELOCITY     X
     1-COORD.        H-TOTAL        H0-INITIAL        PRESSURE  ",//)
 1050 FORMAT("1",//,5X,6HTIME =,F6.3,//)
 1055 FORMAT(1I12,6F15.6)
 1060 FORMAT(////)
 1065 FORMAT("&*dD")
  100 RETURN
      END
C*****

&KYMA  T=00003 IS ON CR00011 USING 00168 BLKS R=0000

C*************************************************
C*****
      SUBROUTINE TOUT(LUIN,NPOIN,IT,IPT), N.C.CHU
C*****
      COMMON/BLOC1/ T0,T,TMAX,IFREQ,IBC,NUMBC,IB(2)
      COMMON/BLOC3/ H(120),V(120),DH(120),DV(120),PP(120)
```

```
      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
      COMMON/BLOC5/ N,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
      COMMON/BLOC6/ HTN(151),VTN(151),HT1(151),VT1(151),TIM(151)
C*****
      WRITE(1,1040)
      WRITE(1,1025) NPOIN,NPLOT
      WRITE(1,1030) IT
      WRITE(1,1010)
 1010 FORMAT("**************************************************************"
     1     ,/,"**** DO YOU WANT PRINTER OUTPUT OF THE NUMBER OF       ****"
     2     ,/,"**** POINTS,PLOTS,ETC.? TYPE 1 FOR YES, 0 FOR NO : _")
      READ(LUIN,*) JPRIN
   25 IF(JPRIN.LT.1) GO TO 30
      WRITE(6,1025) NPOIN,NPLOT
      WRITE(6,1030) IT
      WRITE(6,1035) XL,XL
      WRITE(6,1020) (HT1(I),VT1(I),HTN(I),VTN(I),TIM(I),I=1,IT)
   30 WRITE(1,1015)
 1015 FORMAT("**************************************************************"
     1     ,/,"**** DO YOU WANT TAPE OUTPUT OF ELEVATION AND          ****"
     2     ,/,"**** VELOCITY? TYPE 1 FOR YES, 0 FOR NO : _")
      READ(LUIN,*) JTAPE
   45 IF(JTAPE.LT.1) GO TO 50
      ENDFILE 8
      WRITE(8,1020) (HT1(I),VT1(I),HTN(I),VTN(I),TIM(I),I=1,IT)
      ENDFILE 8
   50 IPT=0
      CALL TPLOT(HTN,TIM,IT,IPT)
      IPT=1
      CALL TPLOT(VTN,TIM,IT,IPT)
      IPT=2
      CALL TPLOT(HT1,TIM,IT,IPT)
      IPT=1
      CALL TPLOT(VT1,TIM,IT,IPT)
C*****
 1020 FORMAT(5X,5F16.6)
 1025 FORMAT(///,10X,"NUMBER OF POINTS, NP = ",I4,///,
     C 10X,"NUMBER OF PLOTS,NPLOT = ",I4)
 1030 FORMAT(10X,///," THE # OF POINTS FOR THE TIME PLOT IS =",I4,////)
 1035 FORMAT(13X,"ELEVATION",7X,"VELOCITY",7X,"ELEVATION",8X,"VELOCITY"
     1        ,10X,"TIME",/,12X,"AT X=0.00",7X,"AT X=0.00",6X,"AT X=",
     2         F5.1,6X,"AT X=",F5.1//)
 1040 FORMAT("ξ*dD")
C*****
      RETURN
      END
```

&PARTM T=00004 IS ON CR00011 USING 00012 BLKS R=0000

```
C*****
C***********************************************************************
C*****
        SUBROUTINE PARTM(GSM,GF,NP,BV,IB,NBW,NBC), G.A.KERAMIDAS
C*****
C***** SUBROUTINE FOR PARTITIONING OF THE GLOBAL STIFFNESS
C***** MATRIX & FORCE VECTOR BY THE METHOD OF DELETION OF
C***** ROWS & COLUMNS ACCORDING TO THE SPECIFIED
C***** BOUNDARY CONDITIONS.
C***** THE GLOBAL STIFFNESS MATRIX GSM(NP,NBAND) CONTAINS
C***** THE NBAND NON-ZERO DIAGONALS OF THE (NP*NP) MATRIX.
C***** <<WARNING>> PARTM IS USED ONLY FOR (EMA) ARRAYS !
C*****
```

```
      DIMENSION BV(NBC),IB(NBC),ND(15)
      EMA GSM(NP,NBW),GF(NP,1)
C*****
       N3=(NBW-1)/2
       NP4=NP-2*NBC
C*****
      DO 10 I=1,NBW
 10     ND(I)=I
C*****
       ID=0
       DO 200 L=1,NBC
       ID=ID+1
       IP=IB(L)
       BC=BV(L)
C*****
C***** MODIFICATION OF THE GLOBAL STIFFNESS MATRIX
C***** & THE FORCE VECTOR
C*****
       NS=NBW-N3
       MP=IP+NS
       IF(MP.GT.NP) GO TO 160
       DO 150 I=1,NS
       IF(I.EQ.IP) GO TO 150
         J1=IP-I+NS
       GF(I,1)=GF(I,1)-GSM(I,J1)*GF(IP,1)/GSM(IP,NS)
       DO 120 J=1,NS
       IF(J.EQ.IP) GO TO 120
         JJ=J-IP+NS
         IJ=J-I+NS
       GSM(I,IJ)=GSM(I,IJ)-GSM(I,J1)*GSM(IP,JJ)/GSM(IP,NS)
 120   CONTINUE
 150   CONTINUE
       GO TO 185
 160   DO 180 I=1,NS
         IN=I+NP4
       IF(IN.EQ.IP) GO TO 180
         JL=IP-IN+NS
       GF(IN,1)=GF(IN,1)-GSM(IN,JL)*GF(IP,1)/GSM(IP,NS)
       DO 170 J=1,NS
         JN=J+NP4
       IF(JN.EQ.IP) GO TO 170
         IL=JN-IP+NS
         JC=JN-IN+NS
       GSM(IN,JC)=GSM(IN,JC)-GSM(IN,JL)*GSM(IP,IL)/GSM(IP,NS)
 170   CONTINUE
 180   CONTINUE
 185   GF(IP,1)=GSM(IP,NS)*BC
        MP0=IP+NS
       IF(MP0.GT.NP) GO TO 195
       DO 190 I0=1,NS
       IF(I0.EQ.IP) GO TO 190
         I1=IP-I0+NS
         I2=I0-IP+NS
         GSM(IP,I2)=0.0
         GSM(I0,I1)=0.0
 190   CONTINUE
       GO TO 200
 195   DO 196 J0=1,NS
        JN0=J0+NP4
       IF(JN0.EQ.IP) GO TO 196
         J1=IP-JN0+NS
         J2=JN0-IP+NS
```

```
                GSM(IP,J2)=0.0
                GSM(JN0,J1)=0.0
  196       CONTINUE
  200       CONTINUE
            RETURN
            END
```

```
&DCOMP T=00004 IS ON CR00011 USING 00008 BLKS R=0000
```

```
      C*****
            SUBROUTINE DCOMP(S,INT,A,N,MAXN,M1,M3), A.J.BAKER
      C*****
      C***** COMPUTE THE LU DECOMPOSITION OF A NONSYMMETRIC
      C***** BANDED MATRIX STORED IN DIAGONAL FORM BY
      C***** USING PARTIAL PIVOTING.
      C*****
            DIMENSION S(MAXN,M1),A(MAXN,M3),INT(MAXN)
      C*****
               L=M1
            DO 120 I=1,M1
              K2=M1+2-I
            DO 100 J=K2,M3
  100          A(I,J-L)=A(I,J)
               L=L-1
              K2=M3-L
            DO 110 J=K2,M3
  110            A(I,J)=0.0
  120       CONTINUE
               L=M1
            DO 220 K=1,N
                        X=A(K,1)
               I=K
              K2=K+1
            IF(L.LT.N) L=L+1
            IF(L.LT.K2) GO TO 150
            DO 140 J=K2,L
            IF(ABS(A(J,1))-ABS(X)) 140,140,130
  130                   X=A(J,1)
               I=J
  140       CONTINUE
  150          INT(K)=I
            IF(X) 160,230,160
  160       IF(I-K) 170,190,170
  170       DO 180 J=1,M3
                        X=A(K,J)
                 A(K,J)=A(I,J)
  180            A(I,J)=X
  190       IF(L.LT.K2) GO TO 220
            DO 210 J=K2,L
                     X=1.0
            IF(A(K,1).NE.0.0) X=A(J,1)/A(K,1)
               S(K,J-K)=X
            DO 200 JJ=2,M3
  200         A(J,JJ-1)=A(J,JJ)-A(K,JJ)*X
  210          A(J,M3)=0.0
  220       CONTINUE
            RETURN
      C*****
      C***** A ZERO PIVOT HAS BEEN FOUND
      C***** PRINT ERROR MESSAGE AND STOP
      C*****
  230       CONTINUE
```

```
        WRITE(6,600) K,I,K2,L,J,A(K,1),A(J,1),X
  600   FORMAT(//,">>>>> ZERO PIVOT IN DCOMP <<<<<",//,
     1   10X,"  K  ,  I  , K2  ,  L  ,  J  ,  A(K,1)  ,  A(J,1)  ,
     2     X",//,10X,5I5,3F10.4)
        STOP
        END
```

&SOLVE T=00004 IS ON CR00011 USING 00006 BLKS R=0000

```
C*****
C*************************************************************************
C*****
        SUBROUTINE SOLVE(S,INT,F,A,N,MAXN,M1,M3), A.J.BAKER
C*****
C***** COMPUTE THE SOLUTION OF THE LINEAR SYSTEM
C***** USING THE LU DECOMPOSITION OF THE STIFFNESS
C***** MATRIX PROVIDED BY DCOMP AND THE RIGHT
C***** HAND SIDE VECTOR.
C*****
        DIMENSION S(MAXN,M1),F(MAXN),A(MAXN,M3),INT(MAXN)
C*****
           L=M1
        DO 130 K=1,N
           I=INT(K)
        IF(I-K) 100,110,100
  100         X=F(K)
           F(K)=F(I)
           F(I)=X
  110     K2=K+1
        IF(L.LT.N) L=L+1
        IF(L.LT.K2) GO TO 130
        DO 120 I=K2,L
              X=S(K,I-K)
  120      F(I)=F(I)-X*F(K)
  130   CONTINUE
           L=1
        DO 180 II=1,N
           I=N+1-II
              X=F(I)
           M=I-1
        IF(L-1) 140,160,140
  140   DO 150 K=2,L
  150         X=X-A(I,K)*F(K+M)
  160      F(I)=X/A(I,1)
        IF(L-M3) 170,180,180
  170      L=L+1
  180   CONTINUE
        RETURN
        END
```